P517611.PDF [Page: 3 of 42]

Defence Research and Development Canada
Recherche et développement pour la défense Canada

DEFENCE R&D DÉFENSE

Comparaison des principaux systèmes cryptographiques à clés publiques

J. Rostand

R & D pour la défense Canada
MÉMORANDUM TECHNIQUE
DREO TM 2001-096
Octobre 2001

Canada

517611.PDF [Page: 4 of 42]

Comparaison des principaux systèmes cryptographiques à clés publiques

J Rostand

Centre de recherches pour la défense, Ottawa
Mémorandum technique
DREO TM 2001-096
Octobre 2001

Auteur

J. Rostand, Ph.D.

Approbation scientifique et technique donnée par

M McIntyre
Chef de section / opérations d'information

Autorisation de publier donnée par

G. Marwood
Président du comité de révision des documents

© Sa majesté la reine, représentée par le ministre de la Défense nationale, 2001
© Her Majesty the Queen as represented by the Minister of National Defence, 2001.

105-03241/
517611.PDF [Page: 6 of 42]

Résumé

Les techniques de cryptographie à clés publiques sont largement utilisées aujourd'hui pour la protection de données électroniques. Toutes les techniques n'offrent pas le même niveau de sécurité ou la même performance. Certaines méthodes offrent des avantages marqués sur d'autres. Le but de ce document est de faire le point sur les principales techniques de cryptographie à clés publiques utilisées aujourd'hui. Ces dernières incluent en particulier le système Rivest-Shamir-Adleman (RSA) et la cryptographie sur courbes elliptiques (ECC).

Nous commencerons par un bref rappel du système RSA. Nous comparerons ensuite la sécurité des trois principaux systèmes à clés publiques et terminerons par une analyse de leur efficacité sous les aspect de largeur de bande, d'espace de stockage et de vitesse d'exécution.

Abstract

Public key cryptography is widely used nowadays for electronic data protection. From one particular technique to another, the level of security provided and the efficiency of a practical application may vary greatly. Some methods offer clear advantages over others in particular circumstances. The objective of this document is to describe the current state of the main public key techniques used today. Those techniques include the Rivest-Shamir-Adleman (RSA) system and the elliptic curve cryptography (ECC).

We will start with a short reminder of the RSA technique. Then, we will conduct a comparison of the security offered by the three main public key cryptosystems. We will finish by an analysis of the efficiency of those systems with respect to their bandwith and storage needs and their speed performance.

page intentionnellement laissée en blanc

P517611.PDF [Page: 8 of 42]

Sommaire

Les communications et les transactions électroniques se font de plus en plus nombreuses, particulièrement depuis l'avènement de l'ère commerciale d'Internet. Dans ce contexte, la cryptographie est essentielle pour mettre en place des systèmes d'authentification efficaces et pour obtenir une protection adéquate des données échangées

Les systèmes cryptographiques à clés publiques les plus utilisés aujourd'hui sont ceux basés sur le problème de factorisation comme Rivest-Shamir-Adleman (RSA), ceux basés sur le problème du logarithme discret dans les groupes de corps finis et ceux basés sur le problème du logarithme discret sur les courbes elliptiques. La cryptographie sur courbes elliptiques (ECC - *Elliptic curve cryptography*) est le plus récent de ces systèmes et sans doute aussi le plus prometteur.

Une comparaison minutieuse des trois principaux systèmes à clés publiques montre que ECC offre une meilleure sécurité par bit de clé que les systèmes classiques. Cela est dû a la plus grande difficulté du problème mathématique sous-jacent En effet, le problème du logarithme discret sur les courbes elliptiques demande un temps exponentiel pour être résolu alors que les problèmes de factorisation et de logarithme discret dans les groupes de corps finis ne demandent qu'un temps sous-exponentiel. Ainsi, les protocoles basés sur ECC offrent des avantages en ce qui concerne la grandeur des clés devant être utilisées et par conséquent, des avantages quant à la quantité requise de bande passante et d'espace de stockage. La cryptographie sur courbe elliptiques est aussi performante du point de vue de la vitesse d'exécution. Elle se compare avantageusement à ses rivales sur ce point Ces constations font de ECC un outil tout à fait désigné pour offrir des services cryptographiques dans des environnements de tous genres, mais plus particulièrement dans des environnements restreints comme les téléphones portables, les téléavertisseurs et les cartes à puce

J Rostand, Comparaison des principaux systèmes cryptographiques à clés publiques, TM 2001-096, CRDO, 2001

P517611.PDF [Page: 9 of 42]

Executive Summary

Communications and electronic transactions are becoming more and more numerous, especially since the beginning of the commercial Internet era In that context, cryptography is essential for having efficient authentication systems and good data protection

The most frequently used public key cryptosystems nowadays are based on the factorisation problem as for the Rivest-Shamir-Adleman (RSA) system, on the discrete logarithm problem in groups of finite fields, and on the discrete logarithm problem on elliptic curves. Elliptic curve cryptography (ECC) is the most recent of these techniques and certainly the most promising too

A carefull comparison of the three most popular cryptosystems shows that ECC offers a better security by key bit. That is a consequence of the greater difficulty of the underlying mathematical problem Indeed, the discrete logarithm problem on elliptic curves can be solved in exponential time as opposed to the problems of factorisation and discrete log in groups of finite fields that require a sub-expontential time Therefore, protocols based on ECC offer advantages with respect to the required bandwith and storage. Elliptic curve cryptography is also efficient from the point of view of execution speed Those facts make ECC an appropriate tool for offering cryptographic services in constrained devices such as cell phones, pagers, smart cards, or any other low processing power devices

J Rostand, Comparaison des principaux systèmes cryptographiques à clés publiques, TM 2001-096, DREO, 2001.

P517611.PDF [Page: 10 of 42]

Table des matières

Résumé i

Abstract i

Sommaire iii

Executive Summary iv

Liste des figures vii

Liste des tableaux viii

1 Introduction 1

2 Le système cryptographique RSA 2

2.1 Signature électronique RSA 2

2.2 Cryptage RSA 3

3 Les principaux systèmes à clés publiques 5

4 Comparaison de la sécurité 7

4.1 Temps d'exécution des meilleurs algorithmes 7

4.2 Seuil de sécurité 8

4.3 RSA et le PF 9

4.4 ECC et le PLDCE 10

4.5 Comparaison de RSA et ECC 12

5 Comparaison de l'efficacité 16

5.1 Bande passante 17

5.2 Espace de stockage 18

5.3 Vitesse 20

P517611.PDF [Page: 11 of 42]

6 Conclusion 23

Références 24

Liste des abréviations, acronymes et sigles 26

P517611.PDF [Page: 12 of 42]

Liste des figures

1 Seuil de sécurité SS en fonction de l'année y 13

2 Longueur de clé minimale pour RSA en fonction de l'année 14

3 Longueur de clé minimale pour ECC en fonction de l'année 14

4 Équivalence des longueurs de clé de RSA et ECC 15

P517611.PDF [Page: 13 of 42]

Liste des tableaux

1 Puissance de calcul requise par le NFS 9

2 Puissance de calcul requise par la méthode rhô de Pollard 11

3 Équivalence des longueurs de clé de RSA et ECC 15

4 Longueur d'une signature 17

5 Longueur d'un cryptogramme 18

6 Quantité d'information devant être stockée localement 20

7 Temps d'exécution de primitives cryptographiques 22

8 Temps d'exécution d'un envoi et d'une réception de courriel 22

1 Introduction

Les bases de la cryptographie à clés publiques ont été jetées en 1976 par Diffie et Hellman dans leur article *New directions in cryptography* [1] Depuis, plusieurs systèmes ont été developpés. Les techniques classiques comme Ruvest-Shamir-Adleman (RSA) [2], ElGamal [3] et le *Digital Signature Algorithm* (DSA) [4] sont largement utilisées aujourd'hui. D'autres techniques connaissent, depuis quelques années, une popularité grandissante. Notons en particulier la cryptographie sur courbes elliptiques (ECC – *Elliptic Curve Cryptography*) dont les fondations remontent à 1987 avec la publication de l'article *Elliptic curve cryptosystems* de Koblitz [5] Le déploiement des systèmes cryptographiques sur courbes elliptiques (SCCE) s'est fait plus lentement que celui des systèmes classiques, en particulier parce que les mathématiques sous-jacentes sont plus sophistiquées que pour ces derniers

Le but de ce rapport est de dresser un portrait objectif de la situation des principaux systèmes à clés publiques. y compris ECC Les aspects considérés seront ceux de la sécurité (section 4) et de l'efficacité (section 5). ce dernier incluant la vitesse d'exécution. l'espace de stockage et la bande passante requise

Nous prendrons pour acquis les notions de bases des systèmes à logarithmes discrets classiques et ceux sur courbes elliptiques. Une révision détaillée sur le logarithme discret et les courbes elliptiques est faite dans [6]. Dans un but de complétude, nous débutons, à la prochaine section, par un bref rappel du système RSA.

2 Le système cryptographique RSA

Un des protocoles cryptographiques les plus populaires est RSA. Le système RSA a été inventé en 1977 par Rivest, Shamir et Adleman [2]. Il s'agit d'un système à clés publiques offrant des services de confidentialité et de signature électronique Sa sécurité est basée sur le problème de factorisation.

Dans le but de comparer les principaux systèmes de cryptographie à clés publiques dont RSA fait partie. nous allons maintenant faire un bref rappel de la mécanique de base de ce dernier.

2.1 Signature électronique RSA

But : Un individu A veut signer un message m. Tout individu B devrait être en mesure de vérifier la signature de A, mais ne devrait pas pouvoir forger une signature à la place de A

Dans l'algorithme suivant, A devra se choisir un entier n Nous allons supposer qu'on dispose d'une fonction de hachage (*hash function*) $h : M \to \mathbb{Z}_n$, où M est l'ensemble des messages possibles.

L'algorithme de signature électronique RSA se compose de trois parties · la construction des clés de A, la signature et la vérification.

- **Construction de la clé publique et de la clé privée de A.**

  1) Choisir deux nombres premiers p et q et poser $n := pq$.

  2) Choisir un entier aléatoire e tel que

     - $3 \leq e \leq n-1$ et
     - $\text{pgcd}(e, (p-1)(q-1)) - 1$.

  3) Calculer $d .= e^{-1} \bmod (p-1)(q-1)$.

Clé publique de A· $(n.e)$,	Clé privée de A d

- **Signature du message m par A.**

  1) Calculer $s := h(m)^d \bmod n$

Signature de A pour le message m: s

- **Vérification de la signature par B.**

  1) Calculer $v_1 := h(m)$.

  2) Calculer $v_2 := s^e \bmod n$

La signature est acceptée si et seulement si $v_1 = v_2$.

On vérifie facilement que si A et B suivent l'algorithme ci-haut, alors $v_1 = v_2$. En effet, on a

$$v_2 = s^e = h(m)^{de} = h(m) = v_1 \bmod n.$$

Pour démontrer ces égalités, on utilise l'hypothèse que $de = 1 \bmod (p-1)(q-1)$ et on applique ensuite le petit théorème de Fermat

D'un autre côté, un adversaire voulant forger la signature de A sur un message m doit trouver s tel que $s^e = h(m) \bmod n$. Autrement dit, il doit extraire une racine e-ième de $h(m)$ dans $\mathbb{Z}_n$. Ce problème s'appelle le *problème RSA*. Si B connaît la factorisation de n (c'est-à-dire p et q), alors il peut, à partir de la clé publique de A, facilement calculer d et suivre l'algorithme ci-haut pour obtenir s. Ainsi, le problème RSA n'est pas plus difficile que le problème de factorisation (PF) La plupart des experts croient en fait que ces deux problèmes sont équivalents. Cette équivalence n'a toutefois jamais été démontrée. Il est donc encore possible, en théorie, qu'il existe une technique permettant de résoudre le problème RSA sans avoir à factoriser l'entier n Des découvertes dans cette direction pourraient anéantir totalement le système RSA.

2.2 Cryptage RSA

But : Un individu B veut envoyer un message crypté à un autre individu A. La contrainte est que ce message ne puisse être décrypté que par A Il pourra ainsi être envoyé via un milieu hostile en toute confidentialité.

La construction des clés privé et publique de A se fait comme dans le cas du protocole de signature RSA. Ici, on fait l'hypothèse que le message m qui doit être crypté est un élément de $\mathbb{Z}_n$.

- **Construction de la clé publique et de la clé privée de A.**

  1) Choisir deux premiers p et q et poser $n := pq$

  2) Choisir un entier aléatoire e tel que

     - $3 \leq e \leq n-1$ et
     - $\text{pgcd}(e,(p-1)(q-1)) = 1$.

  3) Calculer $d := e^{-1} \bmod (p-1)(q-1)$.

Clé publique de A: (n,e), Clé privée de A: d

- **Cryptage du message m par B.**

  1) Calculer $c := m^e \bmod n$

Message crypté: c

- **Décryptage du message c par A.**

  1) Calculer $c^d \bmod n$.

$m = c^d \bmod n$.

On vérifie facilement (voir le cas de la signature) l'énoncé du dernier encadré. Un adversaire ne connaissant pas la clé privée de A et voulant retrouver le message m à partir de c et de la clé publique de A est confronté au problème RSA. Comme mentionné précédemment, ce problème est demeuré jusqu'à ce jour aussi difficile que celui de la factorisation de n.

P517611.PDF [Page: 18 of 42]

3 Les principaux systèmes à clés publiques

Il existe une grande variété de schémas à clés publiques Notons, entres autres, les algorithmes de cryptage RSA, Rabin, ElGamal, ECC, McEliece, Merkle-Hellman knapsack, Chor-Rivest knapsack, Goldwasser-Micali probabilistic et Blum-Goldwasser probabilistic et les algorithmes de signature RSA, Rabin, Fiat-Shamir, Guillou-Quisquater, DSA, ElGamal, ECC et Schnorr Pour nos fins de comparaison, nous allons considérer les trois systèmes cryptographiques les plus utilisées à ce jour

NOTATION On note par $\mathbb{F}_q$ le corps fini à q éléments et par $\mathbb{F}_q^*$ le groupe multiplicatif de ce corps. L'ensemble des points d'une courbe elliptique E sur un corps fini $\mathbb{F}_q$ est noté $E(\mathbb{F}_q)$. Pour plus de détails sur ces concepts algébriques, le lecteur est invité à consulter [6]

- **RSA** Le système RSA (Rivest-Shamir-Adleman) est sans aucun doute le plus connu des trois (voir section 2) Sa sécurité est basé sur le problème de factorisation La clé publique est constituée du produit n de deux premiers p et q et d'un entier e appelé l'exposant public. La clé privée est l'inverse de e modulo $(p-1)(q-1)$.

- **DLS** Les systèmes basés sur le problème du logarithme discret dans le corps fini $\mathbb{F}_q^*$ incluent le protocole de cryptage ElGamal, le protocole de signature DSA et celui d'échange de clés Diffie-Hellman Ils ont comme clé publique deux éléments P et Q du groupe $\mathbb{F}_q^*$ La clé privée est le plus petit entier positif n tel que $P^n = Q$

- **ECC** Les systèmes de cette catégorie sont basés sur le problème du logarithme discret sur les courbes elliptiques On y retrouve entre autres les protocoles ECDSA, ECIES et ECDHS (voir [6, section 5]) Ils ont comme clé publique deux points P et Q d'un même sous-groupe cyclique de la courbe elliptique $E(\mathbb{F}_q)$ La clé privée est le plus petit entier positif n tel que $nP = Q$.

La comparaison entre différents schémas de cryptographie à clés publiques se fait selon deux critères distincts la *sécurité* et l'*efficacité*.

La sécurité d'un système se mesure à l'effort qu'il faut déployer pour le briser. Cet effort est en général une combinaison de ressources matérielles et financières, de connaissances et de temps. À la prochaine section, nous répondrons à la question suivante quelle est la puissance de calcul nécessaire pour briser une primitive cryptographique donnée ? Soit dit en passant, on doit toujours garder à l'esprit que les plus grandes menaces sur un système cryptographique sont d'un tout autre ordre les employés qui dévoilent leurs clés secrètes ou qui laissent leurs postes de travail sans protection pendant plusieurs minutes, les mauvaises mises en application des protocoles, etc...

P517611.PDF [Page: 19 of 42]

L'efficacité d'un schéma cryptographique est composée de plusieurs facteurs. Notons, entre autres, la bande passante nécessaire à sa mise en application, l'espace mémoire nécessaire pour l'exécution des calculs et pour le stockage des clés privées et publiques et la vitesse d'exécutions. Nous regarderons ces aspects à la section 5.

4 Comparaison de la sécurité

4.1 Temps d'exécution des meilleurs algorithmes

Chaque système de cryptographie à clés publiques base sa sécurité sur la difficulté à résoudre un problème mathématique. Plus ce problème est difficile à résoudre, plus le système possède une grande sécurité. Voici les problèmes mathématiques sur lesquels reposent les trois principaux systèmes à clés publiques :

- RSA: le problème de factorisation d'entiers (PF)
- DLS: le problème du logarithme discret dans $\mathbb{F}_q^*$ (PLD).
- ECC: le problème du logarithme discret sur une courbe elliptique (PLDCE)

Ces problèmes sont considérés difficiles aujourd'hui parce que les meilleures méthodes connues pour les résoudre sont peu efficaces. Ces méthodes ont des temps d'exécution sous-exponentiels dans les cas du PF et du PLD et exponentiel dans le cas du PLDCE. Plus précisément, le temps d'exécution des meilleurs algorithmes résolvant le PF (*general number field sieve*) ou le PLD (*index-calculus algorithm*) est

$$L_n[1/3, c] := ae^{(c+o(1))(\ln n)^{1/3}(\ln \ln n)^{2/3}},$$

où a et c sont des constantes et où n est le nombre à factoriser dans le cas du PF et l'ordre du groupe $\mathbb{F}_q^*$ dans le cas du PLD. Dans le cas du PLDCE, le temps d'exécution de l'algorithme rhô de Pollard est

$$\frac{\sqrt{\pi n}}{2} + O(\ln n),$$

où n est l'ordre du point P (la base dans laquelle on veut calculer le logarithme discret).

Afin de comparer concrètement l'efficacité des algorithmes permettant de résoudre le PF et le PLDCE, et donc la sécurité des primitives cryptographiques sous-jacentes, il faut mesurer le temps d'exécution de chacun sur un problème donné. Le *mips-an* est une unité de mesure fréquemment utilisée dans la littérature. Elle correspond à la puissance de calcul d'un DEC VAX 11/780 utilisé pendant un an. Cet ordinateur pouvait effectuer approximativement un million d'instructions par seconde (mips), d'où la notation. Cette unité de mesure est loin de faire l'unanimité. Elle fait l'objet de beaucoup de critiques, en particulier en ce qui concerne la définition d'une instruction de base puisque chaque microprocesseur offre un ensemble différent d'instructions. Quoi qu'il en soit, nous utiliserons le mips-an (MA) dans la suite et l'abréviation MMA pour un million de mips-an. À titre indicatif, un processeur Pentium II 450 Mhz utilisé pendant un an fournit un effort équivalent à 450 MA.

On retrouve dans [7] un des premiers modèles mathématiques rigoureux permettant d'évaluer la sécurité des principales primitives cryptographiques utilisées aujourd'hui. Le modèle permet de calculer la longueur des clés nécessaires pour atteindre un niveau de sécurité donné. Toutes les hypothèses sont clairement établies et comme la plupart apparaissent sous forme de paramètres dans le modèle, elles peuvent être modifiées au gré des besoins de tous et chacun

Nous présentons dans les prochains paragraphes les grandes lignes de ce modèle dans les cas du PF et du PLDCE Noter que le PLD offre une difficulté semblable à celle du PF Nous ne le traiterons pas explicitement.

4.2 Seuil de sécurité

La première étape est d'établir le niveau de sécurité que l'on désire obtenir avec notre système cryptographique. Comme on l'a vu ci-haut, celui-ci peut-être mesuré en mips-an. Si par exemple, on choisit comme seuil de sécurité 10^{10} MA, c'est qu'on considère alors que tous les problèmes demandant un effort de calcul supérieur à 10^{10} MA sont impossibles à résoudre ou demandent un effort démesurément grand par rapport à la valeur de l'information protégée par le système cryptographique. Évidemment, le niveau de sécurité que l'on choisit est dépendant du temps. En 1977, 10^5 MA était considéré impossible à réaliser Aujourd'hui, cet effort de calcul prendrait moins de 6 mois sur un réseau de 150 ordinateurs personnels. Afin de tenir compte des améliorations matérielles qui surviennent au fil du temps, le seuil de sécurité SS sera une fonction de l'année y. Ainsi, $SS(y)$ sera le nombre de mips-an que l'on considère comme irréalisable ou suffisamment sécuritaire jusqu'au plus tard en l'an y.

La formule utilisée par Lenstra et Verheul pour calculer le seuil de sécurité est la suivante :

$$SS(y) := \left(5 \times 10^5\right)\left(2^{(y-s)\,12/m}\right)\left(2^{t(y-s)/b}\right) \quad \text{MA}.$$

Dans cette formule, le coefficient 5×10^5 est le nombre de mips-an nécessaires pour briser le *Data Encryption Standard* (DES). Le paramètre s est l'année à laquelle la sécurité qu'offrait DES était considérée comme suffisante. DES a été créé en 1977. La valeur par défaut de s suggérée dans le modèle est 1982. Ainsi, avec cette valeur, le seuil de sécurité en l'an y donnée par la formule est équivalent à la sécurité qu'offrait DES en 1982. Le terme $2^{(y-s)\,12/m}$ est dû à l'augmentation de la vitesse des microprocesseurs entre les années y et s. L'hypothèse est que la vitesse et la mémoire des ordinateurs doublent à tous les m mois. Selon la loi de Moore, m vaut 18 mois. C'est donc la valeur par défaut proposée par les auteurs du modèle. Finalement, le terme $2^{t(y-s)/b}$ est le reflet de l'augmentation du budget d'un éventuel attaquant. Le paramètre t peut prendre les valeurs 1 ou 0 tout dépendant si on veut tenir compte ou pas de ce facteur. Lorsque

$t = 0$, les ressources disponibles de l'ennemi doublent simplement tous les m mois Lorsque $t = 1$, non seulement la vitesse et la mémoire doublent-elles tous les m mois (pour le même coût), mais, de plus, le budget disponible pour l'ennemi double tous les b ans. Les valeurs par défaut suggérées sont $t = 1$ et $b = 10$ ans. Ainsi, par défaut, le seuil de sécurité en l'an y est

$$SS(y) = 500\,000 \times 2^{23(y-1982)/30} \quad \text{MA}.$$

En particulier, on calcul qu'en 2002, pour obtenir un niveau de sécurité semblable à ce qu'offrait DES en 1982, il faut s'assurer que notre système cryptographique ne puisse être brisé en moins de $SS(2002) = 2.1 \times 10^{10}$ MA.

4.3 RSA et le PF

Le meilleur algorithme à ce jour pour factoriser un entier n est le *number field sieve* (NFS). Son temps asymptotique d'exécution est

$$L_n[1/3, (64/9)^{1/3}] = a \exp\left(\left((64/9)^{1/3} + o(1)\right)(\ln n)^{1/3}(\ln \ln n)^{2/3}\right).$$

Une hypothèse courante est celle d'ignorer le terme $o(1)$. On obtient donc

$$NFS(n) := a \exp\left(\left((64/9)^{1/3}\right)(\ln n)^{1/3}(\ln \ln n)^{2/3}\right)$$

Pour estimer la valeur de la constante a, on utilise des données expérimentales obtenues dans le passé.

Le tableau 1 donne la puissance de calcul qui fut requise pour résoudre certains problèmes de factorisation à l'aide de la méthode du NFS. Les problèmes en question sont des défis proposés par la compagnie *RSA Laboratories* et qui consistent à trouver les facteurs premiers d'entiers donnés Ils portent des noms de la forme RSA-x, où x est le nombre de chiffres que contient l'entier à factoriser. Le tableau présente le nombre de bits de chaque entier, la date de sa factorisation, la puissance de calcul qui fut déployée pour résoudre le problème et une référence aux auteurs de la solution La dernière colonne

Problème	Nb bits	Résolu en	mips-an	Référence	Estimation de a
RSA-130	430	avril 1996	1000	[8]	1.6×10^{-15}
RSA-140	463	février 1999	2000	[9]	8.0×10^{-16}
RSA-155	512	août 1999	8000	[10]	4.6×10^{-16}

TAB 1 – Puissance de calcul requise pour résoudre des problèmes de factorisation à l'aide du NFS

est l'estimation de la constante a dans l'expression $NFS(n)$. Pour déterminer cette constante a, on a effectué, dans le cas de la dernière ligne par exemple, le calcul suivant

$$a = \frac{8000}{\exp\left(((64/9)^{1/3})\,(\ln 2^{430})^{1/3}(\ln\ln 2^{430})^{2/3}\right)} = 1.6 \times 10^{-15}.$$

On remarque que les valeurs estimées pour a ont diminué au fil du temps. Cela peut s'expliquer par de légères améliorations apportées à l'algorithme et par le fait que le terme $o(1)$ dans la formule asymptotique est ignoré dans les calculs mais sans doute non nul dans la réalité. Ainsi, les prévisions obtenues par la formule

$$NFS(n) = 1.6 \times 10^{-15} \times e^{(64/9)^{1/3}(\ln n)^{1/3}(\ln\ln n)^{2/3}} \quad \text{MA}$$

surestiment légèrement le temps d'exécution de l'algorithme.

La longueur de clé minimale pour RSA proposée par Lenstra et Verheul dans leur modèle est le plus petit entier k tel que

$$NFS(2^k) \geq SS(y)\left(2^{(y-1999)\,12/r}\right).$$

Cet entier k sera noté $RSA(y)$. Ainsi, avec une clé de $RSA(y)$ bits, le temps d'exécution de l'algorithme NFS est supérieur ou égal au seuil de sécurité en l'an y ajusté du facteur $2^{(y-1999)\,12/r}$. Ce facteur permet de tenir compte des avancements techniques dans la résolution du problème de factorisation. L'hypothèse est que l'effort de calcul requis pour factoriser un nombre n est diminué de moitié à chaque r mois. Le calcul commence en l'an 1999 puisque la dernière ligne du tableau 1 a été utilisée pour l'estimation de $NFS(n)$. La valeur par défaut du paramètre r est 18 mois. Cette dernière est en accord avec les progrès des 25 dernières années ayant eu une influence sur le problème de factorisation.

4.4 ECC et le PLDCE

L'algorithme le plus performant pour résoudre le problème du logarithme discret sur des courbes elliptiques est l'algorithme rhô de Pollard. Le nombre d'additions (dans le groupe de la courbe elliptique) qui doit être effectuées en moyenne pour extraire le logarithme discret en base P est

$$\frac{\sqrt{\pi n}}{2} + \mathrm{O}(\ln n),$$

où n est l'ordre du sous-groupe engendré par P. Le temps d'exécution d'une addition $P_0 + P_1$ sur une courbe elliptique est proportionnel à la longueur de P_0 et P_1 et donc,

approximativement proportionnel au carré du nombre de bits de P. Ainsi, le temps d'exécution de l'algorithme de Pollard est de la forme

$$Pol(2^k) := a2^{k/2}k^2$$

où k est le nombre de bits de l'ordre du point de base P et où a est une constante

Le tableau 2 nous donne les efforts de calculs qui furent requis ou qui sont estimés pour résoudre certains PLDCE. Les défis ECC ont été proposés par la compagnie *Certicom Corporation*. Il s'agit dans chaque cas d'extraire le logarithme discret d'un point d'une courbe elliptique. Les défis ECC2-x, ECC2K-τ et ECCp-x font intervenir respectivement des courbes elliptiques sur $\mathbb{F}_{2^x}$, des courbes de Koblitz sur $\mathbb{F}_{2^x}$ (voir [6, section 6.3]) et des courbes elliptiques sur $\mathbb{Z}_x$. Pour chaque type de problème, le tableau présente un cas résolu et un cas non résolu. La deuxième colonne du tableau représente le nombre de bits de l'ordre de la base du logarithme. Les colonnes Additions et MMA donnent respectivement le nombre d'additions sur courbes elliptiques et le nombre de millions de mips-an nécessaires pour résoudre les problèmes. Ces données nous permettent d'estimer la constante a de la fonction Pol. On procède essentiellement comme dans le paragraphe précédent. Par exemple, dans le cas de la dernière ligne on trouve

$$a = \frac{2.2 \times 10^6}{2^{109/2}109^2} = 7.27 \times 10^{-15}.$$

Lenstra et Verheul utilisent les données estimées du problème ECCp-109 dans leur modèle. Le temps d'exécution de l'algorithme de Pollard est donc estimé à

$$Pol(2^k) = 7.3 \times 10^{-15}2^{k/2}k^2 \quad \text{MA}$$

pour une clé de k bits. Ainsi, la longueur de clé recommandée pour les systèmes sur courbes elliptiques est la valeur minimale de k satisfaisant l'inégalité

$$Pol(2^k) \geq SS(y)\left(2^{(y-1999)\ 12/c}\right).$$

Problème	Bits	Résolu en	Additions	MMA	Estimation de a
ECC2-97	97	sept. 1999	1.2×10^{11}	0.016	4.3×10^{-15}
ECC2-109	109		1.6×10^{16}	8	2.6×10^{-14}
ECC2K-108	108	avril 2000	2.3×10^{15}	0.4	1.9×10^{-15}
ECC2K-130	130		2.0×10^{18}	540	8.7×10^{-16}
ECCp-97	97	mars 1998	2.0×10^{14}	0.001	5.3×10^{-16}
ECCp-109	109		2.1×10^{16}	2.2	7.3×10^{-15}

TAB. 2 – Puissance de calcul requise (ou estimée) pour résoudre le PLDCE à l'aide de la méthode rhô de Pollard. Voir [11, 12].

P517611.PDF [Page: 25 of 42]

Cette valeur est notée $ECC(k)$. Le facteur $2^{(y-1999)\,12/c}$ représente l'amélioration des techniques de résolution du PLDCE depuis 1999. À tous les c mois, la puissance nécessaire pour résoudre un problème donné est réduite de moitié. Les progrès cryptanalytiques passés reliés au PLDCE sont pratiquement inexistants. La valeur de c qui correspond à ce fait est $c = \infty$ (aucun progrès). Il n'est pas facile de savoir si cette tendance se poursuivra dans le futur. À l'autre extrême, une approche très conservatrice serait de choisir $c = 18$ mois. Nous présenterons ci-après les deux cas. La réalité sera très certainement quelque part entre les deux

4.5 Comparaison de RSA et ECC

En appliquant le modèle de Lenstra et Verheul avec les paramètres $s = 1982$, $m = 18$, $t = 1$, $b = 10$, $r = 18$ et $c = 18, \infty$ on trouve les graphiques 1, 2, 3 et 4. La première figure donne le seuil de sécurité associé à une année y. Ce seuil est mesuré en mips-an et représente une quantité de calculs jugée impossible à réaliser d'ici l'an y. Notez que l'échelle des abscisses est logarithmique Basée sur ces données, la figure 2 montre la longueur de clé minimale du système RSA offrant une sécurité suffisante jusqu'en l'an y La figure 3 est l'équivalent pour les systèmes sur courbes elliptiques. On y présente en rouge l'hypothèse selon laquelle aucun progrès cryptanalytique majeur ne survient ($c = \infty$) et en bleu l'hypothèse selon laquelle les progrès permettent de doubler l'efficacité de résolution du PLDCE à tous les $c = 18$ mois. Finalement, la figure 4 montre les longueurs de clés de RSA et ECC qui sont équivalentes du point de vue de la sécurité Le tableau 3 exhibe quelques valeurs particulières des graphiques précédents.

P517611.PDF [Page: 26 of 42]

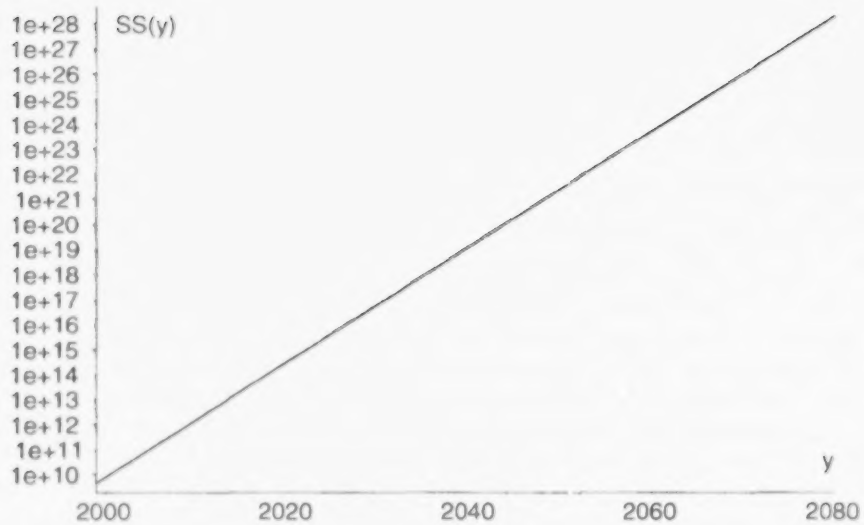

FIG 1 – Seuil de sécurité SS (mips-an) en fonction de l'année y.

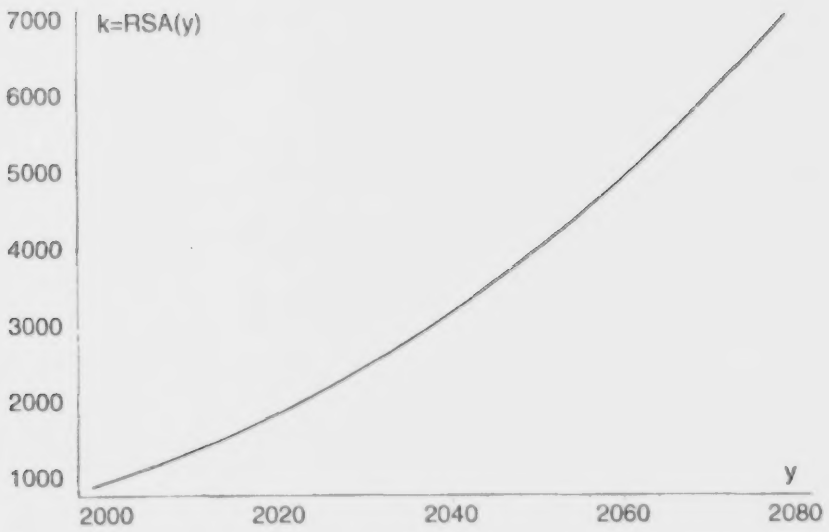

FIG 2 – Longueur de clé minimale (en bits) pour le système RSA en fonction de l'année

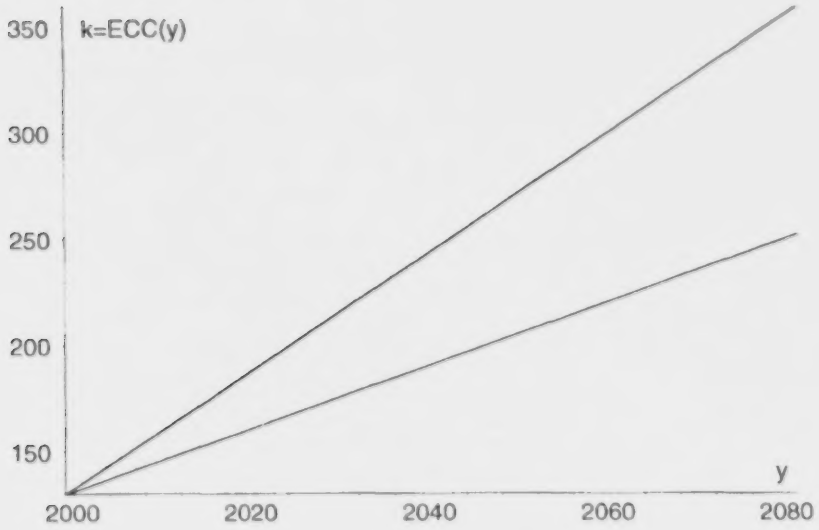

FIG 3 – Longueur de clé minimale (en bits) pour les systèmes ECC en fonction de l'année. Rouge : $c = \infty$, bleu : $c = 18$

P517611.PDF [Page: 28 of 42]

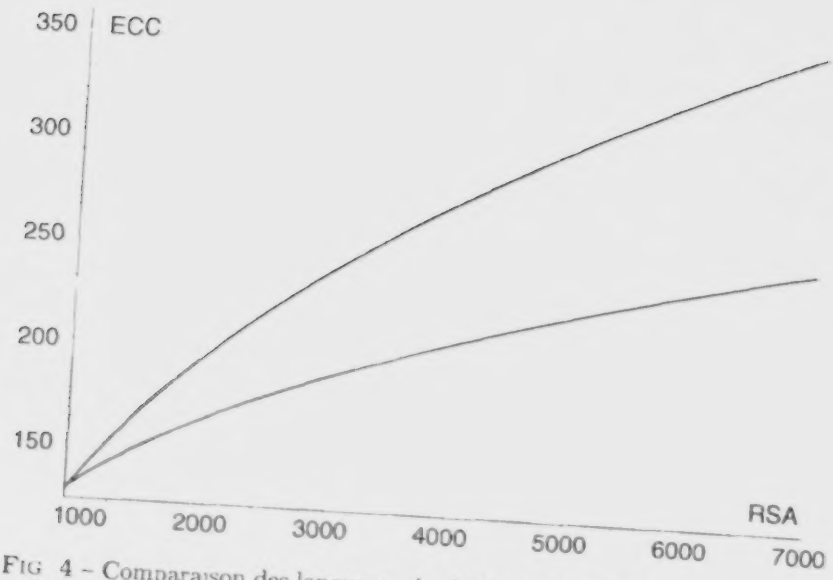

FIG 4 – Comparaison des longueurs de clé (en bits) de RSA et ECC offrant un même niveau de sécurité. Rouge : $c = \infty$, bleu : $c = 18$

Année	SS	RSA	ECC $(c = \infty)$	ECC $(c = 18)$
2000	7.1×10^9	913	131	133
2001	1.2×10^{10}	950	133	135
2002	2.1×10^{10}	988	134	138
2003	3.5×10^{10}	1026	136	141
2004	6.0×10^{10}	1066	137	144
2005	1.0×10^{11}	1106	139	146
2010	1.4×10^{12}	1321	146	160
2015	2.1×10^{13}	1560	153	174
2020	2.9×10^{14}	1822	161	188
2025	4.2×10^{15}	2110	168	202
2030	6.0×10^{16}	2424	176	216
2040	1.2×10^{19}	3133	190	244
2050	2.5×10^{21}	3954	205	272
2100	8.6×10^{32}	9964	280	413

TAB 3 – Comparaison des longueurs de clés de RSA et ECC offrant une sécurité semblable.

P517611.PDF [Page: 29 of 42]

5 Comparaison de l'efficacité

La sécurité d'un système cryptographique est évidemment essentielle pour que celui-ci soit d'un intérêt quelconque À l'étape de l'implémentation concrète d'un protocole cryptographique il faut aussi tenir compte de l'efficacité de celui-ci Les ressources physiques (mémoire, bande passante, puissance de calcul, etc) sont limitées. Dans certains environnements, ces limites ne sont pas un problème. Par exemple, effectuer une signature électronique RSA (avec une clé de 1024 bits) sur un Pentium III (400 Mhz) demande une faible proportion des capacités de l'ordinateur. Les calculs sont complétés en un temps presqu'imperceptible par l'utilisateur Toutefois, la même signature dans un milieu à capacités restreintes, comme par exemple une carte à puce ou une application sans fil, demande une grande proportion des ressources et beaucoup plus de temps. Dans ce contexte, il devient essentiel de comparer l'efficacité des systèmes RSA, DLS et ECC offrant une sécurité semblable.

Une nouvelle variable intervient dans l'étude de l'efficacité d'un système En effet, il ne suffit plus de connaître le problème mathématique sous-jacent à un schéma cryptographique pour obtenir un portrait adéquat de la situation Il faut tenir compte du protocole qui est mis en place et aussi de la manière dont cette implémentation est faite Il existe, pour chaque système cryptographique, plusieurs façons de réaliser une signature, un cryptage ou un échange de clés Il existe d'ailleurs des dizaines d'organisations faisant chacune la promotion de leurs normes en matière de cryptographie à clés publiques (voir [6, section 7]). Nous avons choisi dans cette section quelques protocoles bien établis pour effectués nos comparaisons.

Les aspects de l'efficacité que nous examinerons dans la suite sont : la bande passante, l'espace de stockage et la vitesse d'exécution.

Pour les fins de comparaison, les hypothèses suivantes sont faites

- **RSA**. Le produit $n = pq$ est de 1024 bits L'exposant public e est égal à $2^{16} + 1$ et il est donc de 17 bits
- **DLS**. L'ordre du groupe $\mathbb{F}_q^*$ est de 1024 bits.
- **ECC** L'ordre du point de base de la courbe elliptique est de 160 bits

Ces choix sont largement utilisés en pratique. Notons toutefois que pour ces valeurs, ECC offre une meilleure sécurité que les deux autres systèmes (voir, par exemple, le tableau 3).

5.1 Bande passante

La bande passante requise par un protocole cryptographique est la quantité d'information (mesurée en bits) qui doit être échangée entre les deux parties impliquées pour mener à terme le protocole. Pour comparer les systèmes qui nous intéressent, nous allons évaluer la grandeur de leurs signatures et de leurs messages cryptés. La quantité de bits échangés entre deux parties dépend directement de ces données.

- **Longueur des signatures électroniques**

  Nous allons comparer les protocoles suivants :

  - **RSA** : RSASSA tel que spécifié dans la norme PKCS #1 v.2.1 (voir [13]).
  - **DLS** : DLSP-DSA tel que spécifié dans la norme IEEE P1363 / D13 (voir [14]).
  - **ECC** : ECDSA tel que spécifié dans la norme SEC 1 (voir [15]).

  L'algorithme RSASSA (*RSA Signature Scheme with Appendix*) produit une signature dont la longueur est la même que celle de n. Dans notre cas, il s'agit de 1024 bits. La procédure DLSP-DSA (*Discrete Logarithm Signature Protocol - Digital Signature Algorithm*) est basée sur le problème du log discret dans un sous groupe de $\mathbb{F}_q^*$ d'ordre r. Nous supposerons que r est un entier de 160 bits. L'algorithme produit une signature qui est composée d'une paire de nombres du même ordre que r. La signature a donc une longueur totale de 320 bits. Finalement, ECDSA (*Elliptic Curve Cryptography Signature Algorithm*) donne, comme dans le cas précédant, une signature dont la longueur est deux fois celle du point de base. Ayant choisi un point de base de 160 bits, ECDSA fournit une signature de 320 bits. Ces faits sont résumés dans le tableau 4. Notons que la longueur de la signature n'est pas influencée par la longueur du message à signer m, puisque chacun de ces protocoles applique d'abord au message m une fonction de hachage, ce qui réduit m à quelques centaines de bits tout au plus.

- **Longueur des cryptogrammes**

  Les méthodes à clés publiques ne sont pas, en pratique, utilisées pour crypter de

RSASSA	DLSP-DSA	ECDSA
1024	320	320

TAB 4: Longueur (en bits) d'une signature.

RSAES-OAEP	ElGamal	ECIES
1024	2048	556

TAB. 5 – Longueur (en bits) du résultat du cryptage d'un message de 128 bits.

longs messages m. Les algorithmes à clés symétriques (clés privées) étant beaucoup plus efficaces dans ce domaine, ce sont eux qui sont utilisées sur m pour produire le cryptogramme m' Les systèmes à clés publiques servent, quant à eux, à transmettre le clé symétrique K permettant de decrypter m'. Ainsi, les messages qui doivent être cryptés par des techniques asymétriques sont d'ordinaire relativement courts On supposera ici que la longueur de ces messages est de 128 bits. Cette longueur de clé privée est plus que suffisante en date d'aujourd'hui. Elle offre la même sécurité qu'une clé RSA de 6652 bits et qu'une clé ECC de 345 bits (avec $c = 18$, voir section 4) Pour plus de détail sur cette correspondance, le lecteur est référé à [7].

Le tableau 5 donne les longueurs d'un message de 128 bits crypté par RSAES-OAEP (*RSA Encryption Scheme – Optimal Asymmetric Encryption Padding*, [13]), ElGamal ([16, page 294]) et ECIES (*Elliptic Curve Integrated Encryption Scheme*, [15]).

Les tableaux 4 et 5 permettent de conclure qu'en général, les protocoles de type ECC requièrent une plus petite bande passante que les protocoles de type RSA.

5.2 Espace de stockage

La quantité d'espace requis pour stocker les clés privées et publiques utilisées par un protocole cryptographique peut s'avérer importante dans les environnements où celle-ci est limitée. En conséquence, dans une comparaison des principaux protocoles en clés publiques, on se doit d'évaluer les longueurs des objets suivants :

- La clé privée.
- La clé publique.
- Les paramètres du système
- La signature d'une autorité de certification (AC) sur la clé publique et les paramètres du système.

Dans un certificat complet, d'autres données (nom, adresse électronique, etc) devraient être jointes à la clé publique et aux paramètres du système avant d'être signées par une AC. La longueur de la signature n'est pas modifiées par l'ajout de ces données (on suppose que les schémas de signature utilisent une fonction de hachage. Quant aux données comme telles, l'espace requis pour leur stockage est indépendant du système cryptographique choisi. Il n'est donc pas étudié davantage ici.

- **RSA**

  Dans le système RSA, la clé publique est constituée d'un entier n de 1024 bits et d'un exposant public $e = 2^{16} + 1$, selon nos hypothèses. Ainsi, prenant pour acquis que tous les utilisateurs du système utilisent le même exposant public, la clé publique peut-être stockée en 1024 bits. Si l'exposant public était variable, alors la longueur de clé serait en général de 2048 bits. La clé privée est quant à elle un entier d de 1024 bits (l'inverse de e modulo $(p-1)(q-1)$) Au mieux, la clé privée pourrait être représentée par un entier de 512 bits (l'un des facteurs de n), mais en pratique cela n'est pas utilisé, car le calcul de d à partir de p ou q et de la clé publique est coûteux. Le système RSA n'a besoin d'aucun paramètre autre que ceux déjà inclus dans les clés privées et publiques. Finalement, la signature de l'AC est de 1024 bits (voir section 5 1).

- **DLS**

  Les systèmes à logarithmes discrets sont utilisés dans le contexte d'un groupe et d'un générateur de ce groupe Pour stocker ces informations, 1024 + 1024 = 2048 bits sont nécessaires. Dans le cas des algorithmes de type DSA, il faut aussi spécifier le sous-groupe dans lequel on travaille. Un autre 160 bits est alors nécessaire. Nous ferons donc la distinction entre les systèmes de type DSA et ceux de type ElGamal Au total, c'est 2208 bits qui sont requis pour stocker les paramètres du systèmes dans le cas DSA et 2048 dans le cas ElGamal La clé privée est un entier de 1024 bits dans le cas ElGamal et un entier de 160 bits dans le cas DSA. La clé publique est quant à elle un élément du groupe qui peut donc être représenté par une séquence de 1024 bits. En supposant que dans les deux cas, le système DSA est utilisé par l'AC pour signer le certificat publique, on a besoin pour stocker cette signature d'un espace de 320 bits

- **ECC**

  Le système ECC requiert une courbe elliptique définie sur un groupe fini Pour identifier le groupe, on a besoin d'un entier de 160 bits Pour la courbe elliptique, il faut deux éléments du groupe a et b, un point de la courbe (point de base), et deux entiers (l'ordre n de ce point et le cofacteur $h := |E|/n$) Pour a et b, 160 bits chacun sont nécessaires Pour le point de la courbe elliptique 161 bits est requis si on utilise la technique de compression des points Enfin, les entiers n et h deman-

	RSA	DSA	ElGamal	ECC
Clé privée	1024	160	1024	161
Clé publique	1024	1024	1024	160
Paramètres du système	0	2208	2048	961
Signature de l'AC	1024	320	320	320
Total	3072	3712	4416	1602

TAB. 6 – Quantité (en bits) d'information devant être stockée localement.

dent 160 bits chacun. Ainsi, $160 + 2 \times 160 + 161 + 2 \times 160 = 961$ bits sont requis pour représenter les paramètres du système. La clé privée est un entier de 160 bits La clé publique est un point de la courbe et peut donc être représentée à l'aide de 161 bits. La signature de l'AC fait 320 bits.

Toutes ces constatations sont compilées dans le tableau 6 Il est clair que les systèmes ECC requierent un espace de stockage plus petit que ceux des autres systèmes.

5.3 Vitesse

Un autre aspect incontestablement important lorsqu'on compare plusieurs systèmes cryptographiques est la vitesse à laquelle les différentes opérations de base (signature, vérification, cryptage et décryptage) sont effectuées. Entre un millième de seconde et une demi-seconde, il y a peu de différence pour l'utilisateur Mais entre une et dix secondes, la différence est gigantesque.

La vitesse d'exécution d'un algorithme cryptographique est très dépendante de la manière dont celui-ci est implémenté. Ainsi, ce qu'on obtient en général est le temps d'exécution d'une implémentation particulière d'un algorithme Lorsqu'on veut comparer des systèmes cryptographiques comme RSA, DLS ou ECC, ces données sont certes utiles mais ne disent pas tout. Il faut garder cette remarque en tête lorsqu'on observe les temps d'exécution présentés ci-après.

Il existe dans la littérature beaucoup d'études ayant porté sur le temps d'exécution des algorithmes cryptographiques qui nous intéressent. On trouve dans [17] une liste de résultats à propos de la cryptographie sur courbes elliptiques Brown, Cheung, Hankerson, Hernandez, Kirkup et Menezes ont comparé les temps d'exécution de RSA, DSA, ECDSA, ECIES et ElGamal sur trois plate-formes différentes Nous reproduisons ici les résultats qu'ils ont trouvés. Pour plus de détails, voir [18].

Les algorithmes ont été implémentés via la norme de sécurité PGP (*Pretty Good Privacy*,

voir [19, 20]) sur les trois plate-formes suivantes.

- **RIM**. Le téléavertisseur bidirectionnel *Blackberry* de *Research In Motion* (RIM) Cet assistant numérique personnel est construit sur un processeur Intel 386 10 Mhz. Il dispose de 2 Mb de mémoire flash et de 304 Kb de SRAM
- **Palm**. Le *PalmPilot* est muni d'un processeur Motorola 68000 16 Mhz de type *Dragonball* Il dispose de 2 à 4 Mb de mémoire en plus de sa mémoire morte
- **PC**. Un Pentium II 400 Mhz.

Le tableau 7 donne les résultats obtenus Les temps d'exécution sont en millisecondes. On remarque d'abord que chacun des algorithmes étudiés donne une performance acceptable sur un PC Toutefois, dans un environnement restreint comme le téléavertisseur de RIM et le PalmPilot, il en est tout autrement.

Les premières lignes du tableau montrent que le système RSA est beaucoup plus efficace lors des opérations publiques (cryptage et vérification) que lors des opérations privées (décryptage et signature). Dans le cas de RIM, par exemple, le facteur est presque 15. Une opération privée prend environ 15.9 secondes alors qu'une opération publique prend environ 1.1 seconde. Pour un utilisateur du téléavertisseur, un temps d'attente de 15 secondes pour chaque opération privée est inacceptable

Dans le cas de ECC, ce sont les opérations privées qui sont les plus rapides. Elles requièrent en moyenne 1.72 secondes alors que les opérations publiques demandent 3.2 secondes. Même si ces dernières sont plus lentes que dans le cas RSA, leur temps moyen d'exécution est tout de même tolérable.

À titre d'exemple pratique, considérons le cas où deux individus veulent s'échanger un courriel à l'aide de leur téléavertisseur Blackberry de RIM. Lors de l'envoi, le courriel doit d'abord être signé et ensuite crypté par l'auteur Puis, à la réception, il doit être décrypté et par la suite la signature doit être vérifiée par le récepteur. Le tableau 8 permet de constater qu'un envoi ou une réception de courriel utilisant ECC est approximativement trois fois et demie plus rapide qu'en utilisant le système RSA Il est clair que cela constitue un avantage appréciable pour l'utilisateur.

Protocole	Opération	RIM	Palm	PC
RSA	Construction des clés	580 405	1 705 442	2 740.87
RSA	Cryptage	1 241	2 670	5.34
RSA	Décryptage	15 901	36 284	67 32
RSA	Signature	15 889	36 130	66 56
RSA	Vérification	1 008	2 374	3 86
ECC	Construction des clés	1 085	1 891	2.12
ECIES	Cryptage	3 132	5 458	6.67
ECIES	Décryptage	2 114	3 564	4.69
ECDSA	Signature	1 335	2 230	2.64
ECDSA	Vérification	3 243	5 370	6.46
ElGamal	Construction des clés	—	—	1 200 157
ElGamal	Cryptage	26 588	73 978	67.78
ElGamal	Décryptage	57 248	148 059	144.73
DSA	Construction des clés	—	—	54 674
DSA	Signature	9 529	25 525	24 28
DSA	Vérification	18 566	52 286	47.23

TAB 7 – Les temps d'exécution sont en millisecondes. RSA utilise 1024 bits et $e = 2^{16} + 1$. ECC utilise une courbe elliptique aléatoire sur $\mathbb{F}_{2^{163}}$. DSA et ElGamal sont en 1024 bits.

Protocole	Opérations	Temps total
RSA	Envoi : cryptage et signature	17 130
RSA	Réception : décryptage et vérification	16 909
ECC	Envoi : cryptage et signature	4 467
ECC	Réception : décryptage et vérification	5 357

TAB 8 – Temps d'exécution (millisecondes) d'un envoi et d'une réception de courriel sur le téléavertisseur de RIM.

6 Conclusion

Nous avons présenté dans ce document une comparaison des principaux systèmes cryptographiques à clés publiques, en l'occurrence le système RSA, les systèmes basés sur les logarithmes discrets dans $\mathbb{F}_q^*$ et les systèmes sur courbes elliptiques. La comparaison a porté sur la sécurité de ces systèmes et de leur efficacité.

L'étude sur la sécurité de RSA et ECC a montré que la difficulté des problèmes mathématiques sous-jacents à ces systèmes n'est pas la même. Le problème du logarithme discret sur courbes elliptiques est plus difficile à résoudre que le problème de factorisation d'entiers. En conséquence, pour un même niveau de sécurité, les longueurs de clés devant être utilisées dans les systèmes ECC sont significativement plus petites que celles qui doivent être utilisés dans RSA. Par exemple, pour l'an 2005, des clés de longueurs 1106 bits et 139 bits pour RSA et ECC respectivement offrent un niveau de sécurité équivalent à ce que donnait DES en 1982.

L'étude de l'efficacité des principaux systèmes à clés publiques a été conduite sous trois aspects : la bande passante, l'espace de stockage et la vitesse d'exécution. Dans les deux premiers, la cryptographie sur courbes elliptiques devance clairement ses rivales. Pour la vitesse d'exécution, le verdict est partagé. Les opérations publiques (vérification et cryptage) sont en général plus rapides (un facteur de l'ordre de 2) dans le système RSA que dans ECC. Cela est dû au petit exposant publique $e = 3$ ou $e = 2^{16} + 1$ utilisé dans plusieurs protocoles basés sur RSA. Par contre, les opérations privées (signature et décryptage) sont nettement plus rapide dans ECC (approximativement 15 fois). Ainsi, lorsqu'on combine, comme cela se fait usuellement, cryptage avec signature et décryptage avec vérification, ECC sort encore une fois gagnante.

Toutes ces différences n'ont que peu d'importance dans des systèmes très rapides et n'ayant pas de sévères contraintes de stockage et de bande passante (par exemple un ordinateur de bureau). Par contre, dans les domaines où l'on fait face à des contraintes de capacité de stockage, de largeur de bande et de vitesse d'exécution, la cryptographie sur courbes elliptiques a des avantages certains sur les systèmes classiques. L'avenir de ECC est donc prometteur, en particulier pour des applications dans les appareils sans fil comme les téléphones cellulaires et les téléavertisseurs.

Références

[1] W Diffie et M. E. Hellman **New directions in cryptography**. *IEEE Transactions on Information Theory*, 22 :644–654, 1976.

[2] R.L. Rivest, A Shamir et L M Adleman. **A method for obtaining digital signatures and public-key cryptosystems** *Communications of the ACM*, 21 :120–126, 1978

[3] T ElGamal **A public key cryptosystem and a signature scheme based on discrete logarithms** *IEEE Transactions on Information Theory*, 31 .469–472, 1985

[4] NIST National Institute of Standards and Technology. **Digital Signature Standard**, FIPS Publication 186, 1994. Disponible sur internet : `http://csrc.nist.gov/publications/fips`

[5] N. Koblitz. **Elliptic curve cryptosystems**. *Mathematics of Computation*, 48 :203–209, 1987.

[6] J. Rostand **Courbes elliptiques et cryptographie** Mémorendum technique, DREO TM 2001-095, Octobre 2001.

[7] Arjen K. Lenstra et Eric R. Verheul. **Selecting Cryptographic Key Sizes**. To appear in The Journal of Cryptology, Springer-Verlag.

[8] James Cowie, Bruce Dodson, R.-Marije Elkenbracht-Huizing, Arjen K Lenstra, Peter L. Montgomery et Joerg Zayer. **A world wide number field sieve factoring record : on to 512 bits**. In Kwangjo Kim et Tsutomu Matsumoto, editors, *Advances in Cryptology - Asiacrypt '96*, volume 1163 of *Lecture Notes in Computer Science*, pages 382–394, Berlin, 1996. Springer-Verlag

[9] Stefania Cavallar, Bruce Dodson, Arjen Lenstra, Paul Leyland, Walter Lioen, Peter L. Montgomery, Brian Murphy, Herman te Riele et Paul Zimmermann **Factorization of RSA-140 using the Number Field Sieve** In Lam Kwok Yan, Eiji Okamoto et Xing Chaoping, editors, *Advances in Cryptology - Asiacrypt '99*, Lecture Notes in Computer Science, Berlin, 1999. Springer-Verlag

[10] Stefania Cavallar, Bruce Dodson, Arjen K Lenstra, Walter Lioen, Peter L Montgomery, Brian Murphy, Herman te Riele, Karen Aardal, Jeff Gilchrist, Gérard Guillerm, Paul Leyland, Joël Marchand, François Morain, Alec Muffett, Chris Putnam, Craig Putnam et Paul Zimmermann **Factorization of a 512-Bit RSA Modulus**. In B. Preneel, editor, *Advances in Cryptology - EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 1–18. Springer-Verlag, mai 2000.

[11] Bruce Schneier. **Elliptic curve public-key cryptography** *Crypto-Gram*, Novembre 1999.
`http://www.counterpane.com/crypto-gram.html`.

[12] Certicom Corp **ECC2K-108 Challenge solved**, Avril 2000.
`http://www.certicom.com/research/news/ecc2k108.html`

P517611.PDF [Page: 38 of 42]

[13] RSA Security Inc **PKCS #1 v2.1 : Public-Key Cryptography Standard**, 5 janvier 2001.

[14] IEEE **IEEE P1363 / D13, Standard Specifications for Public Key Cryptography**, 12 novembre 1999

[15] Certicom Corporation **SEC 1 : Standards for Efficient Cryptography / Elliptic Curve Cryptography Version 1.0**, 20 septembre 2000.

[16] A Menezes, P van Oorschot et S. Vanstone. **Handbook of Applied Cryptography**. CRC Press, Inc., 1997.

[17] R. Dahab et J. Lopez **Performance of elliptic curve cryptosystems** Technical Report IC-00-08, Institute of Computing, University of Campinas, May 2000 `http://www.dcc.unicamp.br/ic-main/publications-e.html`.

[18] M. Brown, D Cheung, D Hankerson, J. Hernandez, M Kirkup et A. Menezes **PGP in constrained wireless devices** In *Proceedings of the 9th USENIX Security Symposium*, 2000

[19] J. Callas **OpenPGP Specification and Sample Code**. Printers Inc Bookstore, Palo Alto, march 1999.

[20] S Garfinkel **PGP : Pretty Good Privacy** O'Reilly & Associates, 1995

P517611.PDF [Page: 39 of 42]

Liste des abréviations, acronymes et sigles

Acronyme	Définition	Page
AC	Autorité de certification	18
DES	*Data Encryption Standard*	8
DLS	*Discrete Logarithm Systems*	5
DLSP-DSA	*Discrete Logarithm Signature Protocol – Digital Signature Algorithm*	17
DSA	*Digital Signature Algorithm*	1
ECC	*Elliptic Curve Cryptography*	1
ECDSA	*Elliptic Curve Cryptography Signature Algorithm*	17
ECIES	*Elliptic Curve Integrated Encryption Scheme*	18
MA	mips-an	7
mips	million d'instructions par seconde	7
MMA	million de mips-an	7
NFS	*number field sieve*	9
PF	Problème de factorisation	3
PGP	*Pretty Good Privacy*	20
PLD	Problème du logarithme discret	7
PLDCE	Problème du logarithme discret sur courbes elliptiques	7
RIM	*Research In Motion*	21
RSA	Rivest-Shamir-Adleman	1
RSAES-OAEP	*RSA Encryption Scheme – Optimal Asymmetric Encryption Padding*	18
RSASSA	*RSA Signature Scheme with Appendix*	17
SCCE	Systèmes cryptographiques sur courbes elliptiques	1

SANS CLASSIFICATION

COTE DE SÉCURITÉ DE LA FORMULE
(plus haut niveau du titre, du résume ou des mots-clefs)

FICHE DE CONTRÔLE DU DOCUMENT

1 PROVENANCE (le nom et l'adresse)	2 COTE DE SÉCURITÉ (y compris les notices d'avertissement, s'il y a lieu)
Centre de Recherches pour la défense Ottawa 3701 avenue Carling Ottawa, Ontario, K1A 0Z4	SANS CLASSIFICATION

3 TITRE (Indiquer la cote de securité au moyen de l'abréviation (S, C, R ou U) mise entre parentheses, immediatement après le titre)
Comparaison des principaux systèmes cryptographiques a clés publiques (U)

4 AUTEURS (Nom de famille, prénom et initiales Indiquer les grades militaires, ex Bleau, Maj Louis E)
Rostand, Jérémie

5 DATE DE PUBLICATION DU DOCUMENT (mois et année)	6a NOMBRE DE PAGES	6b NOMBRE DE REFERENC
Octobre 2001	viii + 26	20

7 DESCRIPTION DU DOCUMENT (La categorie du document, par exemple rapport, note technique ou mémorandum Indiquer les dates lorsque le ra couvre une période définie)
Mémorandum technique

8 PARRAIN (le nom et l'adresse)
Section d'opérations d'information, CRDO, Ottawa

9a NUMÉRO DU PROJET OU DE LA SUBVENTION (Spécifier si c'est un projet ou une subvention)	9b NUMERO DE CONTRAT
5bf27	

10a NUMÉRO DU DOCUMENT DE L'ORGANISME EXPEDITEUR	10b AUTRES NUMÉROS DU DOCUMENT
DREO TM 2001-096	N/A

11 ACCÈS AU DOCUMENT (Toutes les restrictions concernant une diffusion plus ample du document, autres que celles inhérentes à la cote de sécur

- [x] Diffusion illimitee
- [ ] Diffusion limitée aux entrepreneurs des pays suivants (spécifier)
- [ ] Diffusion limitee aux entrepreneurs canadiens (avec une justification)
- [ ] Diffusion limitée aux organismes gouvernementaux (avec une justification)
- [ ] Diffusion limitee aux ministeres de la Defense
- [ ] Autres

12 ANNONCE DU DOCUMENT (Toutes les restrictions à l'annonce bibliographique de ce document Cela correspond, en principe, aux données d' au document (11) Lorsqu'une diffusion supplémentaire (à d'autres organismes que ceux précisés à la case 11) est possible, on pourra élargir le de diffusion de l'annonce)
Annonce complète illimitée

SANS CLASSIFICATION

COTE DE LA SÉCURITÉ DE LA FORMULE
(plus haut niveau du titre, du résumé ou des mots-clefs)

SANS CLASSIFICATION

COTE DE LA SÉCURITÉ DE LA FORMULE
(plus haut niveau du titre, du résumé ou des mots-clefs)

13 SOMMAIRE (Un résumé clair et concis du document Les renseignements peuvent aussi figurer ailleurs dans le document Il est souhaitable sommaire des documents classifiés soit non classifié Il faut inscrire au commencement de chaque paragraphe du sommaire la cote de s applicable aux renseignements qui s'y trouvent, à moins que le document lui-même soit non classifié Se servir des lettres suivantes (S), (C), (U) Il n'est pas nécessaire de fournir ici des sommaires dans les deux langues officielles à moins que le document soit bilingue)

Les techniques de cryptographie à cles publiques sont largement utilisées aujourd'hui pour la protection de donnees électroniques Toutes les tech n'offrent pas le même niveau de sécurité ou la même performance Certaines méthodes offrent des avantages marques sur d'autres Le but de ce doc est de faire le point sur les principales techniques de cryptographie à clés publiques utilisées aujourd'hui Ces dernières incluent en particulier le sy Rivest-Shamir-Adleman (RSA) et la cryptographie sur courbes elliptiques (ECC)

Nous commencerons par un bref rappel du système RSA Nous comparerons ensuite la sécurité des trois principaux systèmes à clés publiq terminerons par une analyse de leur efficacité sous les aspect de largeur de bande, d'espace de stockage et de vitesse d'exécution

14 MOTS-CLÉS, DESCRIPTEURS OU RENSEIGNEMENTS SPÉCIAUX (Expressions ou mots significatifs du point de vue technique, qui caractéri document et peuvent aider à le cataloguer Il faut choisir des termes qui n'exigent pas de cote de sécurité Des renseignements tels que le moc l'équipement, la marque de fabrique, le nom de code du projet militaire, la situation geographique, peuvent servir de mots-clés Si possible, c choisir des mots-clés d'un thésaurus, par exemple le "Thesaurus of Engineering and Scientific Terms (TESTS)" Nommer ce thésaurus Si l'on r pas trouver de termes non classifiés, il faut indiquer la classification de chaque terme comme on le fait avec le titre)

cryptographie à clés publiques, RSA, DSA, ECC, courbes elliptiques, cryptographie, signature électronique, cryptage

Defence R&D Canada
is the national authority for providing
Science and Technology (S&T) leadership
in the advancement and maintenance
of Canada's defence capabilities.

R et D pour la défense Canada
est responsable, au niveau national, pour
les sciences et la technologie (S et T)
au service de l'avancement et du maintien des
capacités de défense du Canada.

#517611
CA020870

DEFENCE R&D DÉFENSE

www.drdc-rddc.dnd.ca